

Zastosowanie metod wyznaczania przepływu w sieciach do planowania manewru wojsk

Z. TARAPATA

e-mail: zbigniew.tarapata@wat.edu.pl

Wojskowa Akademia Techniczna, Wydział Cybernetyki,
ul. Gen. S. Kaliskiego 2, 00-908 Warszawa

W artykule przedstawiono opis sposobu wykorzystania metod wyznaczania przepływu w sieciach do rozwiązania specyficznego problemu planowania manewru wojsk. Zdefiniowano model sieci formalnej, bazującej na danych z cyfrowej mapy terenu, wykorzystywanej jako model środowiska w problemie planowania manewru. Sformułowano optymalizacyjny problem planowania przegrupowania K obiektów z rejonu startowego (reprezentowanego przez podzbiór wierzchołków sieci formalnej) do rejonu docelowego, z dodatkowym ograniczeniem na rozłączność dróg. Opisano sposób modyfikacji sieci pierwotnej oraz poszukiwania jednego z rozwiązań dopuszczalnych sformułowanego problemu planowania przegrupowania z użyciem metody znajdowania przepływu maksymalnego w sieci zmodyfikowanej. Przedyskutowano metodę poszukiwania rozwiązania optymalnego bazującą na algorytmie znajdowania przepływu zaspokajającego o minimalnym koszcie w pewnej sieci zastępczej. Opisane metody zilustrowano przykładami obliczeniowymi. Oszacowano złożoność obliczeniową prezentowanych algorytmów. Artykuł kończy omówienie rozszerzeń sformułowanego problemu wyjściowego oraz metod ich rozwiązywania.

Keywords: planowanie manewru, drogi rozłączne, przepływy w sieciach

1. Wprowadzenie

Problem planowania manewru jest jednym z zasadniczych elementów działań bojowych i jest związany z przegrupowaniem jednostek wojskowych na polu walki zarówno w czasie działań bojowych, jak i na etapie ich planowania. W symulatorach walki, ale również w systemach wspomagania dowodzenia, proces planowania i symulacji przemieszczania obiektów ma wpływ na dokładność, adekwatność, efektywność i inne charakterystyki tych systemów (w szczególności symulatorów). Planowanie i symulacja przegrupowania jednostek są również podstawowymi problemami rozpatrywanymi w systemach typu CGF (ang. *Computer Generated Forces*) [23]. Nieodłączną częścią tego typu systemów są moduły planowania tras przemieszczania bazujące na modelach terenu i wykorzystujące przetworzone z nich informacje. Dla przykładu moduł tego typu stosowany w systemie *ModSAF* opisany został w [19]. W wielu istniejących systemach symulacyjnych oraz systemach wspomagania decyzji stosuje się różne rozwiązania związane z omawianą tematyką [2], [5], [18], [32]. W [2] autorzy opisują system mieszany planowania tras („po drogach-na przełaj”) z użyciem systemów typu GIS w symulatorach. Opisują m.in. tzw. hierarchiczny system planowania

dróg, jako część prognostycznego systemu analiz taktycznych (ang. *Predictive Intelligence Military Tactical Analysis System (PIMTAS)*). Rozszerzeniem omawianego systemu jest tzw. *Route Planning Uncertainty Manager (RPLUM)* przedstawiany w [15]. System ten wykorzystuje analizę niepewności do analizy terenu i projektowania tras przemieszczania wykorzystując różne kategorie niepewności (prawdopodobieństwo, rozmytość). Autor [5] opisuje moduł planowania tras w systemie o nazwie *Close Combat Tactical Trainer (CCTT)*. Kreitzberg w [18] opisuje analizator przemieszczania (ang. *Tactical Movement Analyzer (TMA)*). System ten wykorzystuje mapy cyfrowe, zdjęcia satelitarne, warunki pogodowe i rodzaj pojazdu, aby określić przejezdność kwadratu terenu (o długości boku ok. 0.5 km). W polskim systemie symulacyjnym SSWSO „Złocień”, wykorzystywany jest dwupoziomowy model terenu [33] bazujący na przetworzonych danych z map cyfrowych w formacie VPF (ang. *Vector Product Format*): w postaci siatki kwadratów o regularnych bokach oraz w postaci sieci drogowo-kolejowej. Oba te modele są zintegrowane, tzn. istnieje możliwość przejścia z jednego modelu do drugiego (patrz punkt 2). Wiele z prezentowanych systemów wykorzystuje do planowania przemieszczania tzw. ciągły

algorytm Dijkstry (ang. *Dijkstra's Continuous Algorithm*) zaproponowany przez Mitchell'a [21]. Przegląd możliwych rozwiązań w tym zakresie przedstawiono m.in. w [32].

W niniejszym opracowaniu przedstawiono sposób wykorzystania metod wyznaczania przepływów w sieciach do rozwiązania specyficznego problemu planowania manewru wojsk lądowych. W punkcie 2 przedstawiono model terenu. Punkt 3 zawiera zdefiniowanie specyficznego problemu planowania manewru wojsk. W punkcie 4 przedstawiono metodę poszukiwania jednego z rozwiązań dopuszczalnych zdefiniowanego problemu z wykorzystaniem metody poszukiwania przepływu maksymalnego, a w punkcie 5 – metodę wyznaczania rozwiązania optymalnego z wykorzystaniem metody wyznaczania przepływu najtańszego w sieci. Punkt 6 zawiera opis kilku rozszerzeń rozpatrywanego problemu wyjściowego oraz metod ich rozwiązania. Opracowanie kończą wnioski.

2. Model terenu

Jednym z zasadniczych problemów w trakcie rozwiązywania wszelkiego rodzaju zagadnień transportowych (a zadanie przemieszczania jest szczególnym przypadkiem zadania transportowego), który niejednokrotnie rzutuje na charakter i dokładność rozwiązania, jest odpowiednie odwzorowanie „środowiska”, w którym jest rozpatrywane zagadnienie transportowe. Modelem takiego „środowiska” jest z reguły graf z opisanymi na nim odpowiednimi funkcjami, które opisują „środowisko”. Wówczas graf taki (jako struktura „środowiska”) staje się siecią, która w sposób ilościowy (czasami również jakościowy) opisuje otoczenie. W naszym przypadku opisem „środowiska” może być np. sieć dróg lub inna sieć opisująca możliwości poruszania się w rzeczywistym terenie z opisaną na gałęziach funkcją odległości między wierzchołkami sieci. Dla przykładu, w systemie *JTLS* [27], [36] teren reprezentowany jest za pomocą heksagonów (sześciokątów foremnych) o rozmiarach od 1km do 16km. Podobnie teren reprezentowany jest w systemie *CBS* [8], z tym, że heksagony mają rozmiar 3km. W *CBS* zastosowano dodatkowo reprezentację wektorowo-obszarową terenu. W obu tych systemach stosuje się ręczne i automatyczne metody planowania tras dla obiektów (np. w *CBS* tzw. kontroler wprowadza kolejne punkty (współrzędne), przez które ma przebiegać droga). Z kolei w systemie

„Złocien”, który powstał na Wydziale Cybernetyki WAT na potrzeby SZ RP, wykorzystywany jest model otoczenia $S_o(t) = \langle Z_1(t), Z_2(t) \rangle$ pola walki zawierający dwa składowe modele, powstałe po przetworzeniu danych z cyfrowych map terenu w formacie VPF [33]: pierwszym jest model bazujący na podziale terenu na regularną siatkę kwadratowych podobszarów ($Z_1(t)$), a drugi to model sieci dróg (wraz z mostami, stałymi przeprawami, wiaduktami) - $Z_2(t)$. Pierwszy z modeli ($Z_1(t)$) jest wykorzystywany do modelowania rozmieszczenia wojsk w terenie oraz na jego podstawie (a dokładniej: na podstawie funkcji opisanych na kwadratach sieci) określać będziemy wpływ terenu na przebieg działań (skuteczność ognia, widoczność, zdolność do maskowania itp.), natomiast drugi z modeli ($Z_2(t)$) wykorzystywany jest przy modelowaniu przegrupowania (przemieszczania) kolumn marszowych, dowozu zaopatrzenia, ewakuacji rannych itp., czyli takich działań, dla których wymagana jest sieć drogową. Regularna sieć kwadratów dzieli interesujący nas obszar terenu na kwadratowe podobszary tych samych rozmiarów (o długości boku równej 200 m), przy czym każdy z kwadratów jest jednorodny ze względu na szereg charakterystyk go opisujących (stopień osłabienia prędkości, stopień widoczności, stopień maskowania, itp.). W sieci drogowo-kolejowej wierzchołkami sieci formalnej są skrzyżowania drogowe i kolejowe, a krawędzie (łuki) łączą sąsiadujące skrzyżowania. Oba te modele są zintegrowane, tzn. istnieje możliwość przejścia z jednego modelu do drugiego. Jest to możliwe dlatego, że każdy kwadrat w modelu sieci kwadratów zawiera informacje o odcinkach dróg przez niego przebiegających. Z drugiej strony, każdy odcinek drogi (reprezentowany w modelu sieci drogowo-kolejowej przez łuki) posiada informacje, przez które kwadraty sieci kwadratów przebiega. Dlatego też droga przemieszczania się dowolnego obiektu może zawierać zarówno odcinki sieci drogowo-kolejowej, jaki i kwadraty sieci kwadratów. W obu modelach zakłada się, że funkcje opisane na wierzchołkach i łukach sieci zależą od czasu. Na potrzeby artykułu, dla ustalenia opisu sieci S będącej modelem terenu, przyjmujemy następującą definicję:

$$S = \langle G, D \rangle \quad (1)$$

gdzie: G - graf Berge'a, bez pętli, na którym opisana jest sieć S , $G = \langle W_G, U_G \rangle$, $W = |W_G|$,

$U_G \subset W_G \times W_G$, $U = |U_G|$; $D = [d_{w,w'}]_{W \times W}$ - macierz odległości między wierzchołkami sieci. Niech maksymalne prędkości obiektów manewrowych (pojazdów, grup pojazdów lub oddziałów) wynoszą $v_1, v_2, \dots, v_k, \dots, v_K$, przy czym $\forall_{k=1,K} v_k > 0$. Ponieważ prędkości te mogą być różne dla różnych jednostek, więc minimalny czas $t_{w,w'}$ pokonania łuku $u_{w,w'}$ dla każdego $w, w' \in \{1, \dots, W\}$ jest wektorem K -wymiarowym i ma postać:

$$t_{w,w'} = \langle t_{w,w'}^1, t_{w,w'}^2, \dots, t_{w,w'}^k, \dots, t_{w,w'}^K \rangle \quad (2)$$

gdzie $\forall k = 1, \dots, K$ zachodzi:

$$t_{w,w'}^k = \begin{cases} \frac{d_{w,w'}}{v_k \cdot \zeta(u_{w,w'})}, & u_{w,w'} \in U_G \wedge \zeta(u_{w,w'}) > 0 \\ 0, & w = w' \\ +\infty, & u_{w,w'} \notin U_G \vee \zeta(u_{w,w'}) = 0 \end{cases} \quad (3)$$

Funkcja $\zeta(u_{w,w'})$ przyjmuje wartości z przedziału $[0,1]$ i ich interpretacja jest taka, iż opisują one stopień osłabienia prędkości poruszania się po danym łuku ze względu na warunki topograficzne, atmosferyczne, itp. odcinka drogi (trasy), którego reprezentantem jest łuk (więcej na ten temat można znaleźć np. w [31], [33]). Wobec tego sieć S z (1) może być zdefiniowana jako:

$$S' = \langle G, t_u \rangle \quad (4)$$

gdzie: $t_u = [t_{w,w'}]_{W \times W}$ (5)

Sieć S z (1) będziemy wykorzystywać w przypadku szukania najkrótszych dróg przemieszczania w sensie odległości terenowej natomiast sieci S' z (4) - w przypadku szukania najkrótszych dróg przemieszczania w sensie czasu.

3. Zdefiniowanie problemu manewru wojsk lądowych

Problem manewru wojsk lądowych sprowadza się do zaplanowania przegrupowania K obiektów manewrowych (pojazdów, kolumn pojazdów, pododdziałów) z pewnego rejonu początkowego do pewnego rejonu docelowego (końcowego).

Za rejonny możemy uważać pewne podzbiory wierzchołków sieci S . Obiekty podlegające przegrupowaniu przyjmują pewne ugrupowanie i najczęściej tym ugrupowaniem jest ugrupowanie marszowe, czyli kolumny pojedynczych obiektów. Problem wyznaczania planu przegrupowania dla wielu kolumn jest problemem bardzo złożonym. Zagadnienie to znacznie się komplikuje, z tego względu, że kolumny marszowe są rozciągnięte na pewnej długości, a w związku z tym określone wierzchołki i drogi w pewnych okresach czasu są zajęte przez daną kolumnę. Wobec powyższego w pewnych odcinkach czasu nie będzie możliwości przejazdu przez dany wierzchołek czy odcinek (łuk) drogi. Zajętość ta może wynikać również z działalności jednostek nadrzędnych.

Zaproponowana w dalszej części punktu definicja problemu nie ogranicza go jednakże do problemu planowania manewru wyłącznie kolumn obiektów. Przyjmujemy jedynie założenie, że szerokość obiektu manewrowego jest nie większa niż jego długość. Wówczas, konstruując sieć kwadratów terenu, długość boku najmniejszego umownego kwadratu obszaru jednorodnego topograficznie będzie nie mniejsza niż szerokość tego obiektu manewrowego. Jeżeli obiekty manewrowe mają niejednorodną wielkość to za długość boku najmniejszego umownego kwadratu obszaru jednorodnego topograficznie przyjmujemy szerokość największego (w sensie szerokości) obiektu manewrowego.

Każdy k -ty obiekt manewrowy ma tzw. długość czasową δ_k (jak np. kolumna pojazdów). Wierzchołek sieci będziemy uważali za zajęty z chwilą, gdy czoło obiektu manewrowego „wjedzie” na niego, a za zwolniony - jeżeli ogon obiektu „opuści” go. Czas zajętości wierzchołka równy jest długości czasowej δ_k obiektu manewrowego:

$$\delta_k = \frac{\alpha_k}{v_k} \quad (6)$$

gdzie:

δ_k - długość czasowa k -tego obiektu manewrowego;

α_k - długość obiektu manewrowego;

v_k - prędkość obiektu manewrowego.

Pojęcie zajętości danego wierzchołka powoduje to, że w danej chwili nie może przejeżdżać przez niego więcej niż jeden obiekt. Przyjmujemy również założenie, że odcinek drogi między dwoma wierzchołkami (łuk) jest zajęty od chwili, kiedy czoło obiektu wjedzie na ten

odcinek, aż do momentu, kiedy ogon obiektu opuści go.

W dalszej części opracowania zajmiemy się problemem wyznaczania optymalnego planu przegrupowania (manewru w sensie przemarszu) K obiektów z ustalonego rejonu początkowego do rejonu końcowego. Ustalmy również, że w dalszej części opracowania obiekt manewrowy nazywać będziemy krótko obiektem (chyba, że wyraźnie zaznaczymy, iż chodzi o inny obiekt niż manewrowy).

Pod pojęciem planu przegrupowania (dla k -tego obiektu, $k=1, \dots, K$) będziemy rozumieć uporządkowane zbiory: wierzchołków I_k tworzących drogę oraz momentów T_k (tworzących harmonogram) osiągnięcia wierzchołków należących do danej drogi:

$$I_k = (i^0(k), i^1(k), \dots, i^r(k), \dots, i^{R_k}(k)) \quad (7)$$

$$T_k(I_k) = T_k = (\tau^0(k), \tau^1(k), \dots, \tau^r(k), \dots, \tau^{R_k}(k)) \quad (8)$$

gdzie: $i^r(k) \in W_G$ - r -ty wierzchołek na drodze dla k -tego obiektu; $\tau^r(k)$ - moment osiągnięcia wierzchołka $i^r(k)$ przez czoło k -tego obiektu,

$$\bigvee_{k=1, K} \bigvee_{r=0, R_k-1} \tau^{r+1}(k) \geq \tau^r(k), \quad R_k - \text{liczność}$$

zbioru wierzchołków drogi dla k -tego obiektu.

Pod pojęciem planu przegrupowania $\mathbf{P}(\mathbf{I}, \mathbf{T})$ K obiektów będziemy rozumieć uporządkowany zbiór par:

$$\mathbf{P}(\mathbf{I}, \mathbf{T}) = ((I_1, T_1), (I_2, T_2), \dots, (I_K, T_K)) \quad (9)$$

gdzie:

$$\mathbf{I} = (I_1, I_2, \dots, I_k, \dots, I_K) \quad (10)$$

$$\mathbf{T} = (T_1, T_2, \dots, T_k, \dots, T_K) \quad (11)$$

Oznaczmy przez i^s początkowy rejon (zbiór wierzchołków) rozlokowania K obiektów,

$$i^s = \{i_1^s, i_2^s, \dots, i_k^s, \dots, i_K^s, \dots, i_{K_s}^s\} \quad (12)$$

przy czym $K_s \geq K$.

Jeżeli przez τ_k^s oznaczymy moment rozpoczęcia przegrupowania k -tego obiektu, to uporządkowany zbiór wyżej wymienionych momentów będzie miał postać:

$$\tau^s = (\tau_1^s, \tau_2^s, \dots, \tau_k^s, \dots, \tau_K^s) \quad (13)$$

Rejon docelowy dotarcia K obiektów oznaczymy przez i^d ,

$$i^d = \{i_1^d, i_2^d, \dots, i_k^d, \dots, i_K^d, \dots, i_{K_d}^d\} \quad (14)$$

przy czym $K_d \geq K$.

Założymy również (co wynika z analizy problemu przegrupowania i sensowności konstruowania planu manewru), że $i^s \cap i^d = \emptyset$, tzn. zbiory początkowych i końcowych rejonów rozlokowania są rozłączne.

Przez τ_k^d będziemy oznaczać moment przybycia k -tego obiektu do jednego z wierzchołków rejonu docelowego. Zbiór wymienionych momentów będziemy oznaczać przez :

$$\tau^d = (\tau_1^d, \tau_2^d, \dots, \tau_k^d, \dots, \tau_K^d) \quad (15)$$

przy czym

$$\tau_k^d = \tau_k^s + \sum_{r=1}^{R_k} t_{i^{r-1}(k), i^r(k)}^k + \delta_k, \quad k = \overline{1, K} \quad (16)$$

gdzie:

τ_k^s - element zbioru z (13);

$t_{i^{r-1}(k), i^r(k)}^k$ - czas pokonania (długość

łuku na drodze dla k -tego obiektu, pomiędzy wierzchołkami $i^{r-1}(k)$ oraz $i^r(k)$, (zob. (3));

δ_k - określone przez (6).

Oznaczmy przez M zbiór następującej postaci :

$$M = \{I^1, I^2, \dots, I^l, \dots, I^L\} \quad (17)$$

Jeśli przez (10) oznaczymy dowolny dopuszczalny K -elementowy uporządkowany zbiór dróg w sieci S (przy ustalonych wierzchołkach początkowych i końcowych) natomiast przez (11) zbiór terminów odpowiadających temu zbiorowi dróg, to naszym zadaniem jest tak ułożyć plan przegrupowania, tzn. wyznaczyć takie I^* oraz odpowiadające mu T^* , dla których zachodzi:

$$\tau(I^*, T^*) = \min_{I \in M} \tau(I, T) \quad (18)$$

$$\text{gdzie: } \tau(I, T) = \sum_{k=1}^K (\tau_k^d - \tau_k^s) \quad (19)$$

tzn. chcemy minimalizować sumę czasów przejazdu wszystkich obiektów z rejonu początkowego do docelowego. Czasami może nas interesować funkcja $\tau(I, T)$ inaczej zdefiniowana, tzn.

$$\tau(I, T) = \max_{k=1, K} \tau^{R_k}(k) = \max_{k=1, K} \tau_k^d \quad (20)$$

Przy takiej definicji żądamy (wstawiając (20) do (18)), aby chwila dotarcia najwolniejszego obiektu do rejonu docelowego była jak najwcześniejsza.

Należy zauważyć, że przy tak zdefiniowanym problemie nie wymaga się, aby drogi dla poszczególnych obiektów były parami rozłączne. Celowo tak zdefiniowano problem, gdyż jest to przypadek ogólniejszy. W praktyce jednak często żąda się, aby drogi przemieszczania dla poszczególnych obiektów były parami rozłączne (tzw. drogi niezależne). Ograniczenie to podyktowane jest najczęściej wymogami bezpieczeństwa przegrupowania.

W problemie (18), czy to z funkcją podkryterialną (19), czy też z funkcją (20), możemy stosować dodatkowe ograniczenia, które zawężają nam zbiór rozwiązań dopuszczalnych. Ograniczeniami tymi mogą być, np. odległości między drogami dla sąsiednich obiektów, przejazd przez wybrane wierzchołki pośrednie sieci, itp. (zob. np. w [16], [22], [29]).

Przedstawimy obecnie rozwiązanie problemu (18)-(19) przy spełnieniu warunków niezależności (rozłączności) wyznaczanych dróg (czyli objekty będą przemieszczać się w sieci dróg bezkolizyjnie), tzn. zakładać będziemy, że wierzchołki należące do dróg dla K obiektów (patrz (7)) spełniają zależność:

$$\forall_{k, k'=1, K} \quad \forall_{\substack{r=0, R_k \\ r'=0, R_k \\ r \neq r': k=k'}} \quad i^r(k) \neq i^{r'}(k') \quad (21)$$

W przypadku gęstej sieci tras przejazdu, tak zdefiniowany problem wystarczająco dokładnie opisuje rzeczywisty manewr. W przypadku przegrupowania (marszu) kolumn obiektów po sieci drogowej, stosuje się kilka rozłącznych dróg, co wynika z regulaminów i reguł taktycznych (np. batalion do marszu otrzymuje jedną drogą, brygada – 2 lub 3, dywizja – od 3 do 5, itd.).

4. Metoda wyznaczania jednego z rozwiązań dopuszczalnych

Metodami potencjalnie nadającymi się do rozwiązania problemu (18)-(19) z ograniczeniami (21) dla rozłącznych parami dróg dla K obiektów mogą być algorytmy poszukiwania N -tej najkrótszej drogi zawarte np. w [12], [14], [20] oraz inne ([29], [30]). W celu znalezienia rozwiązania dopuszczalnego powyższego problemu posłużymy się algorytmem wyznaczania maksymalnego przepływu w pewnej sieci S^* , która powstanie w oparciu o sieć S' z (4) (lub S z (1)); na tym etapie nie ma to znaczenia, którą sieć wybierzemy). Wykorzystamy w tym celu

fundamentalne twierdzenie o maksymalnym przepływie i mini-malnym przekroju, którego dowód można znaleźć, m.in. w ([9], str. 500), ([17], str. 224), ([39], str. 171).

Twierdzenie 1

W każdej sieci S wartość dowolnego przepływu maksymalnego z s do t jest równa przepustowości dowolnego przekroju minimalnego.

Dla nas będzie szczególnie interesujący jeden z wniosków, z tego twierdzenia (czasami formułowany jako tzw. twierdzenie Mengera).

Wniosek 1 ([39], str. 172)

Jeżeli przepustowość każdego łuku jest liczbą całkowitą, wtedy sieć można traktować jako digraf DAG , w którym przepustowość opisana na łuku u oznacza krotność połączeń (łukowych) między wierzchołkami łuku u . Wartość przepływu maksymalnego określa wówczas liczbę wszystkich dróg łukowo rozłącznych z wierzchołka s do t w S . ■

Z wniosku płynie natychmiastowa sugestia, co do sposobu znajdowania w sieci liczby dróg rozłącznych łukowo między ustaloną parą wierzchołków. Opiszemy teraz sposób tworzenia sieci S^* , w której będziemy szukać K dróg rozłącznych wierzchołkowo.

Opis tworzenia sieci S^*

1. Przypisujemy wszystkim łukom sieci S wartości przepustowości równe 1 (bo nasza sieć bazuje na grafie Berge'a, w którym dla każdej uporządkowanej pary wierzchołków może istnieć co najwyżej jeden łuk) i wówczas wartość maksymalnego przepływu (zgodnie z wnioskiem 1) między ustaloną parą wierzchołków będzie określać liczbę wszystkich dróg łukowo rozłącznych między tą parą. Nas jednakże interesują drogi rozłączne wierzchołkowo. Wbrew pozorom problem ten jest niewiele trudniejszy od problemu poprzedniego (punkt 2).
2. Tym razem musimy dodatkowo przypisać wierzchołkom naszej sieci przepustowości równe 1. Ale wówczas należy przekształcić sieć, aby pozbyć się przepustowości opisanych na wierzchołkach. Można tego dokonać konstruując graf $G^* = \langle W^*, U^* \rangle$, na którym opisana będzie sieć S^* ([9], str. 503-504): zastępujemy każdy wierzchołek v z S przez dwa wierzchołki v' i v'' w S^* oraz prowadzimy łuk od v' do v'' , któremu przypisujemy wartość przepustowości taką, jaką miał wierzchołek v (czyli 1). Wszystkie łuki, które

„wchodzący” do wierzchołka v - „wchodzić” będą do wierzchołka v' , a te które „wychodzący” z wierzchołka v - będą „wychodzić” z v ”. Przepustowości tych łuków w zmodyfikowanej sieci będą takie, jak w sieci pierwotnej (czyli równe 1).

3. Pozostaje jeszcze jedna modyfikacja sieci S' - należy wyróżnić w sieci S^* tzw. źródło s i odpływ t . Źródło s jest to dodatkowy wierzchołek, który dodajemy do zbioru wierzchołków sieci S^* i z tego wierzchołka wychodzący będą łuki łączące K_s wierzchołków ze zbioru i^s wierzchołków rejonu początkowego (pamiętać należy, że zgodnie z punktem 2 algorytmu, dołączamy te łuki do wierzchołków typu v'). Łukom tym przypisujemy przepustowości równe 1. Odpływ t jest to dodatkowy wierzchołek, który dodajemy do zbioru wierzchołków sieci S^* i do tego wierzchołka będą dochodzić łuki z K_d wierzchołków (opisanych jako v'') zbioru i^d wierzchołków rejonu docelowego. Łukom tym również przypisujemy przepustowości równe 1. Zmodyfikowana sieć S^* (graf G^*) zawiera zatem $V=2W+2$ wierzchołków oraz $A=U+W+K_s'+K_d'$ łuków.

Zdefiniujemy teraz formalnie nasz problem znajdowania K rozłącznych wierzchołkowo dróg w zmodyfikowanej sieci S^* . Niech $S^* = \langle G^*, c_{ij} \rangle$, gdzie c_{ij} oznacza przepustowość łuku (i,j) . Przepływem nazywamy przyporządkowanie nieujemnych liczb f_{ij} łukom (i,j) , spełniające następujące warunki:

1. dla każdego łuku (i,j) w S^* zachodzi:

$$0 \leq f_{ij} \leq c_{ij} \quad (22)$$

2. sieć S^* zawiera wyróżniony wierzchołek s (źródło), dla którego zachodzi:

$$\sum_{i \in W^*} f_{si} - \sum_{i \in W^*} f_{is} = F \quad (23)$$

gdzie w pierwszej sumie sumujemy po wszystkich łukach wchodzących do wierzchołka i a w sumie drugiej - po wszystkich łukach wchodzących do wierzchołka s . Wielkość F nazywa się wartością przepływu.

3. sieć S^* zawiera wyróżniony wierzchołek t (odpływ), dla którego zachodzi:

$$\sum_{i \in W^*} f_{ti} - \sum_{i \in W^*} f_{it} = -F \quad (24)$$

4. dla każdego z pozostałych wierzchołków j (wierzchołki pośrednie) zachodzi:

$$\sum_{i \in W^*} f_{ji} - \sum_{i \in W^*} f_{ij} = 0 \quad (25)$$

Problem maksymalnego przepływu zdefiniujemy teraz następująco: dla danych S^* , s , t , znaleźć funkcję przepływu $[f_{ij}]^*$, spełniającą warunki (22)-(25) taką, że

$$F([f_{ij}]^*) = \max_{[f_{ij}] \in V(s,t)} F([f_{ij}]) \quad (26)$$

gdzie $V(s,t)$ - zbiór wszystkich możliwych przepływów w S^* z s do t .

Problem ten można rozwiązać wykorzystując jeden ze znanych algorytmów wyznaczania maksymalnego przepływu, np. algorytm Forda-Fulkersona [1], [9], ([17], str. 228), [39]. Jeżeli po rozwiązaniu problemu otrzymamy wartość przepływu $F([f_{ij}]^*) < K$ wówczas w sieci S^* (i w konsekwencji w S) nie istnieje K rozłącznych wierzchołkowo dróg z podzbioru wierzchołków i^s do i^d (zbiór rozwiązań dopuszczalnych jest zbiorem pustym). W przeciwnym przypadku istnieje K dróg rozłącznych wierzchołkowo z podzbioru wierzchołków i^s do i^d , a drogi te możemy odczytać z sieci S^* po ostatnim kroku algorytmu wyznaczania maksymalnego przepływu w ten sposób, że startujemy K razy z wierzchołka s i poruszamy się po łukach o wartości przepływu 1 aż dojdziemy do wierzchołka t . Odczytane w ten sposób ciągi naprzemienne wierzchołków i łuków wskazują k -tą drogę z podzbioru i^s do i^d . Innymi słowy, K dróg w S^* odczytujemy w ten sposób, że identyfikujemy tzw. łańcuchy powiększalne ([17], str. 226), ([39], str. 173), które tworzyliśmy na etapie działania algorytmu Forda-Fulkersona. Przy okazji należy zauważyć ważną cechę, która również jest dowodem na to, że jeżeli w sieci S^* istnieje K dróg rozłącznych wierzchołkowo, to zidentyfikowane łańcuchy będą określać nam drogi rozłączne. Otóż z idei konstruowania łańcuchów powiększalnych podczas realizacji algorytmu Forda-Fulkersona wynika, że przy przepustowościach wszystkich łuków w S^* równych $c_{ij}=1$ niemożliwa jest sytuacja, aby przez jakikolwiek łuk (a pamiętajmy, że poprzez zastąpienie każdego wierzchołka v z S przez dwa wierzchołki v' i v'' w S^* , z wierzchołków v z S „zrobiliśmy” łuki $v' \rightarrow v''$ w S^*) przechodziła więcej niż jedna droga (zob. warunek (22) oraz (25)).

Zaprezentowaną wyżej ideę wyznaczania rozwiązania dopuszczalnego problemu manewru przedstawia przykład 1.

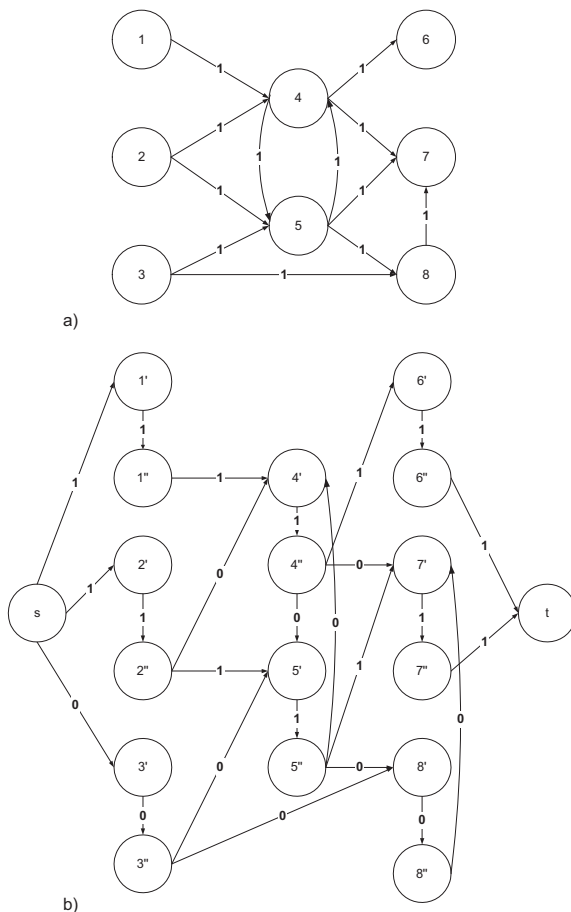
Przykład 1

Chcemy wyznaczyć drogi rozłączne wierzchołkowo dla $K=2$ obiektów, przy czym $i^s=\{1,2,3\}$ oraz $i^d=\{6,7,8\}$. Sieć wyjściowa S znajduje się na Rys.1a.

Na Rys.1b znajduje się sieć S^* (jako modyfikacja sieci S zgodnie z algorytmem tworzenia sieci S^*) wraz z zaznaczoną funkcją przepływu po wykonaniu ostatniego kroku algorytmu

Forda-Fulkersona. Łatwo zauważyć, że przepływ ten spełnia warunki (22)-(25), a jego wartość wynosi $\sum_i f_{si} = f_{s1} + f_{s2} + f_{s3} = 1 + 1 + 0 = 2$,

czyli istnieją $K=2$ drogi rozłączne wierzchołkowo z $i^s=\{1,2,3\}$ do $i^d=\{6,7,8\}$. Aby je odczytać startujemy 2 razy z wierzchołka s i przechodzimy po łukach mających wartość przepływu równą 1.



Rys.1 Sieć S do przykładu 1 wraz z przepustowościami c_{ij} łuków (a) oraz odpowiadająca jej sieć S^* (b) z funkcją przepływu $[f_{ij}]^*$ po realizacji algorytmu wyznaczania maksymalnego przepływu

Otrzymujemy dwa następujące ciągi :

$$s \rightarrow 1' \rightarrow 1'' \rightarrow 4' \rightarrow 4'' \rightarrow 6' \rightarrow 6'' \rightarrow t;$$

$$s \rightarrow 2' \rightarrow 2'' \rightarrow 5' \rightarrow 5'' \rightarrow 7' \rightarrow 7'' \rightarrow t.$$

Stąd drogi przemieszczania dla poszczególnych obiektów są następujące : $1 \rightarrow 4 \rightarrow 6$; $2 \rightarrow 5 \rightarrow 7$. Należy zauważyć, że równie dobrze mogliśmy połączyć wierzchołek $8''$ w S^* z t i wówczas otrzymalibyśmy $K=3$ drogi rozłączne wierzchołkowo. Można było również połączyć wierzchołek s tylko z dwoma wierzchołkami z i^s i wówczas mielibyśmy zdefiniowane dokładnie $K=2$ wierzchołki początkowe (po jednym dla każdego z K obiektów). Wówczas problem przegrupowania z rejonu początkowego do rejonu docelowego $K=2$ obiektów sprowadziłby się do problemu wyznaczenia $K=2$ dróg z ustalonych wierzchołków początkowych do ustalonych wierzchołków końcowych. W pewnych jednak sytuacjach (o których napiszemy w dalszej części) wygodniejsze będzie posługiwanie się rejonami (podzbiorami wierzchołków) dla K obiektów, a nie konkretnymi wierzchołkami. ■

Złożoność podstawowego algorytmu Forda-Fulkersona wynosi $O(Af_{max})$, gdzie f_{max} oznacza wartość maksymalnego przepływu w sieci, a efektywniejsza implementacja tego algorytmu (zwłaszcza w przypadku niecałkowitych wartości przepustowości) podana przez Edmonsa i Karpa [13] ma złożoność $O(VA^2)$.

5. Metoda wyznaczania rozwiązania optymalnego

W poprzednim punkcie pokazaliśmy jak można wyznaczyć jedno z rozwiązań dopuszczalnych problemu (18)-(19) z ograniczeniami (21) przy założeniu rozłączności parami dróg dla K obiektów. Zauważyliśmy wówczas, że jeżeli wartość przepływu $F([f_{ij}]^*) < K$ to problem ten nie ma rozwiązania. W przeciwnym przypadku sieć S^* może zawierać wiele przepływów dopuszczalnych z s do t . Istnieje zatem problem znalezienia najtańszego z przepływów dopuszczalnych w sensie funkcji (19). Możemy wobec tego przyjąć, że nasza sieć S^* zostanie rozszerzona o funkcję łukową t_{ij} z (2)-(3) (otrzymamy sieć S^{**}), czyli będzie postaci:

$$S^{**} = \langle G^*, \psi^1, \{t_{ij}, c_{ij}\} \rangle \quad (27)$$

Ponieważ t_{ij} jest wektorem K -wymiarowym (zob. (2)), więc na początku przyjmijmy, że dla każdego łuku (i,j) zachodzi (zob. (3)):

$$t_{ij}^1 = \dots = t_{ij}^k = \dots = t_{ij}^K \quad (28)$$

Wówczas K -wymiarowy wektor t_{ij} długości czasowych łuku (i,j) możemy traktować jako skalar. Znaczący to tyle, że przemieszczamy K jednorodnych obiektów. Przy algorytmie rozwiązującym problem (18)-(19) z ograniczeniami (21) podamy jak można pozbyć się tego ograniczenia.

Problem znalezienia rozwiązania optymalnego możemy obecnie rozpatrywać jako problem znalezienia najtańszego przepływu o określonej wartości $F=K$ w sieci S^{**} i możemy go zdefiniować następująco: dla danych S^{**} , s , t , K ,

$$\sum_{(i,j) \in U^*} t_{ij} f_{ij} \rightarrow \min \quad (29)$$

przy ograniczeniach:

jak w (22);

jak w (23), ale podstawiamy $F=K$;

jak w (24), ale podstawiamy $F=K$;

jak w (25).

Zauważmy, że u nas f_{ij} może przyjmować tylko dwie wartości: 0 lub 1 i wówczas (25) określać będzie minimum z sumy czasów przejazdu K obiektów z s do t (czyli (19), jeśli tylko $\sum_{k=1, K} \tau_k^s = 0$).

Idea algorytmu rozwiązania tego problemu pochodzi od Busackera i Gowena [3]: najpierw znajdujemy najtańszą drogę z s do t , stosując jeden z algorytmów wyznaczania najkrótszych dróg, np. zawarty w [11], [12], [37], a następnie przesyłamy tą drogą możliwie najwięcej jednostek przepływu. Jeśli osiągnęliśmy już przepływ o wartości K , to kończymy obliczenia. W przeciwnym razie modyfikujemy sieć S^{**} (w sposób opisany poniżej), biorąc pod uwagę dotychczas znaleziony przepływ. W zmodyfikowanej sieci ponownie znajdujemy drogę o najniższym koszcie z s do t i przesyłamy nią możliwie najwięcej. Te dwa podstawowe kroki, polegające na znajdowaniu drogi o najniższym koszcie w sieci i modyfikowaniu sieci dla bieżącego przepływu, są na przemian powtarzane tak długo, aż albo otrzymany zostanie przepływ o docelowej wartości K , albo bieżąca sieć nie zawiera drogi z s do t (oznacza to, że K przekracza wartość maksymalnego przepływu F w sieci S^{**}).

Modyfikacja sieci S^{**} [3], ([10], str. 246)

Sieć zmodyfikowana (powiększająca) \bar{S} względem przepływu $F=[f_{ij}]$ w sieci S^{**} ma taką samą strukturę jak sieć S^{**} , a różni się jedynie zmodyfikowanymi przepustowościami \bar{c}_{ij}

i zmodyfikowanymi kosztami \bar{t}_{ij} , które definiujemy następująco:

- Jeśli łuk (j,i) w sieci ma niezerowy przepływ f_{ji} , to wirtualny (pozorny) łuk (i,j) jest użyteczny w przeciwnym kierunku i ma przepustowość f_{ji} , czyli

$$\bar{c}_{ij} = f_{ji}, \quad \text{jeśli } f_{ji} > 0 \quad (30)$$

- Koszt przyporządkowany temu łukowi jest równy $-t_{ji}$, gdyż łuk (i,j) jest użyteczny tylko po to, by zredukować przepływ (a zatem by zredukować także koszt), czyli

$$\bar{t}_{ij} = -t_{ji}, \quad \text{jeśli } f_{ji} > 0 \quad (31)$$

- Podobnie, jeśli przepływ f_{ij} wzdłuż łuku (i,j) jest mniejszy niż jego przepustowość c_{ij} , to możemy przesłać dodatkowy przepływ tym nienasyconym łukiem, pod warunkiem oczywiście, że przepływ f_{ji} ma wartość zero, zatem

$$\bar{c}_{ij} = c_{ij} - f_{ij}, \quad \text{jeśli } f_{ji} = 0 \text{ i } c_{ij} > f_{ij} \quad (32)$$

- Koszt przesłania dodatkowej jednostki przepływu w tym przypadku jest taki sam jak koszt oryginalny, czyli

$$\bar{t}_{ij} = t_{ij}, \quad \text{jeśli } f_{ji} = 0 \text{ i } c_{ij} > f_{ij} \quad (33)$$

- Wreszcie, ponieważ nasycone łuki nie są użyteczne zgodnie z ich orientacją, przyjmujemy

$$\bar{c}_{ij} = c_{ij} - f_{ij} = 0, \quad \bar{t}_{ij} = \infty, \\ \text{jeśli } f_{ij} = c_{ij} \text{ i } f_{ji} = 0 \quad (34)$$

Zauważmy, że powyższe warunki zapewniają spełnienie zależności $f_{ij} f_{ji} = 0$, bowiem w tym samym czasie łuk nie może być użyty do przesyłania przepływu w obu przeciwnych kierunkach. Pozostałe łuki z S^{**} , czyli łuki z zerowym przepływem, nie ulegają zmianie w zmodyfikowanej sieci \bar{S} . ■

Schemat algorytmu znajdowania najtańszego przepływu można przedstawić w sposób następujący [3], ([10], str. 247). Dokładna postać tego algorytmu znajduje się np. w [3], [9] oraz [10].

Algorytm Busackera-Gowena

$$\bar{S} := S^{**};$$

$$\forall_{(i,j) \in U_G} f_{ij} := 0;$$

WHILE (wartość przepływu $< K$) **AND**
 (istnieje droga z s do t w \bar{S}) **DO**
 znaleźć najkrótszą drogę P z s do t w \bar{S} ;
 IF nie istnieje taka droga **THEN**
 EXIT;
 wysłać dodatkowy przepływ wzdłuż P ,
 aż albo P stanie się nasycona, albo
 wartość przepływu osiągnie K ;
 IF droga P staje się nasycona **THEN**
 utworzyć zmodyfikowaną sieć \bar{S} ;
END WHILE. ■

Następujące twierdzenie mówi o tym, że w taki sposób przeprowadzony algorytm daje rozwiązanie optymalne, tzn. najtańszy przepływ:

Twierdzenie 2 [3]

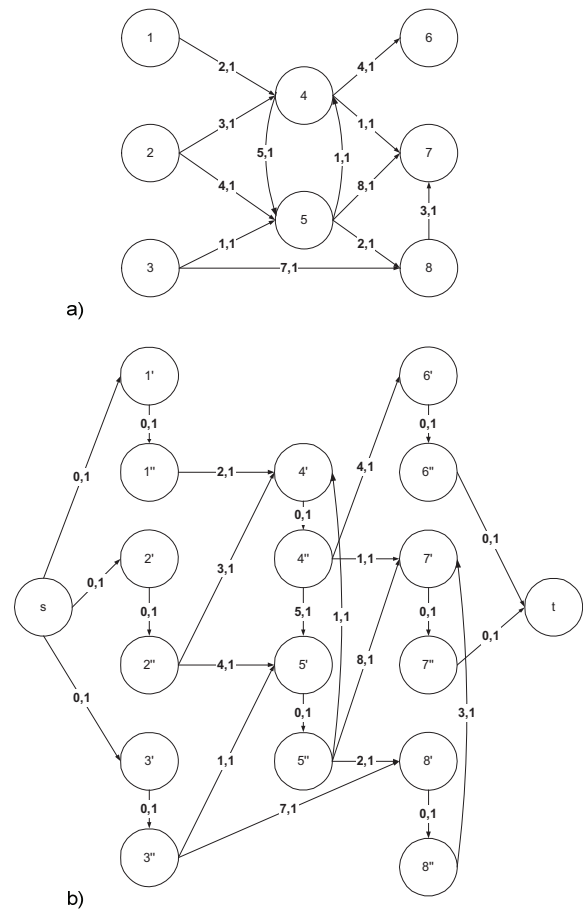
Niech f^1 będzie najtańszym przepływem o wartości F_1 w sieci G , G_1^* - siecią zmodyfikowaną względem przepływu f^1 , a P - najtańszą drogą z s do t w sieci G_1^* . Przepływ f^2 , otrzymany z f^1 poprzez przesłanie dodatkowych jednostek drogą P z s do t , jest także najtańszym przepływem w G o odpowiednio powiększonej wartości. ■

Ideę algorytmu przedstawimy w przykładzie 2.

Przykład 2

Podobnie jak w przykładzie 1, chcemy wyznaczyć drogi rozłączne wierzchołkowo dla dwóch obiektów, przy czym $i^s = \{1,2,3\}$ oraz $i^d = \{6,7,8\}$. Sieć wyjściowa S znajduje się na Rys.2a. Pierwsza liczba z opisanych na łukach podaje koszt (czas) pokonania tego łuku (przyjmujemy założenie, jak w (28)) natomiast druga opisuje przepustowość tego łuku.

Na Rys.2b znajduje się sieć S^{**} (jako modyfikacja sieci S^* z Rys.1b) przy czym pierwsza z liczb na łukach opisuje koszt pokonania łuku, a druga jego przepustowość. Należy zauważyć, że koszt każdego łuku z v' do v'' jest zerowy ponieważ w sieci S , na jej wierzchołkach nie była opisana funkcja kosztu. Z podobnego powodu również zerowy jest koszt na wszystkich łukach (s, v') oraz (v'', t) w S^{**} .



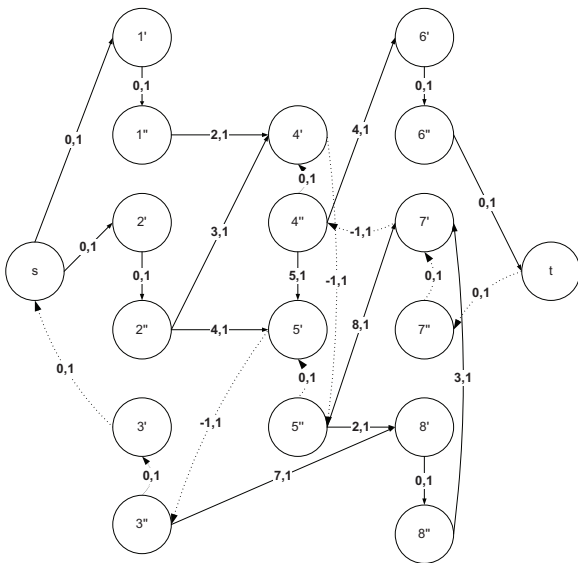
Rys.2. Sieć S do przykładu 2 wraz z kosztami t_{ij} i przepustowościami c_{ij} łuków (a) oraz odpowiadająca jej sieć S^{**} (b)

Iteracja 1

Na początku wszystkim łukom przypisujemy przepływ zerowy. Następnie wyznaczamy najkrótszą drogę z s do t w sieci z Rys.2b; jest nią droga $P = (s, 3', 3'', 5', 5'', 4', 4'', 7', 7'', t)$. Koszt przesłania jednostki przepływu tą drogą jest równy $0+0+1+0+1+0+1+0+0=3$, a maksymalny możliwy przepływ wynosi oczywiście $F=1 < K=2$. Określamy więc przepływ o wartości 1 po tej drodze, czyli $f_{s3'} = f_{3'3''} = f_{3''5'} = f_{5'5''} = f_{5''4'} = f_{4'4''} = f_{4''7'} = f_{7'7''} = f_{7''t} = 1$, modyfikujemy sieć uwzględniając otrzymany przepływ (zob. Rys.3) i przechodzimy do następnej iteracji. Zauważmy, że na Rys.3 nie narysowaliśmy łuków, którymi został przesłany aktualny przepływ, gdyż zgodnie z (34), dla każdego takiego łuku (i,j) zachodzi: $\bar{c}_{ij} = c_{ij} - f_{ij} = 0$ oraz $\bar{t}_{ij} = \infty$ i łuk o takich parametrach nigdy więcej nie będzie brany pod uwagę przy wyznaczaniu drogi najkrótszej, bo jego koszt czasowy jest równy nieskończoności.

Iteracja 2

Najkrótszą drogą z s do t w sieci z Rys.3 jest $P=(s, 1', 1'', 4', 5'', 8', 8'', 7', 4'', 6'', t)$. Koszt przesłania jednostki przepływu tą drogą jest równy $0+0+2+(-1)+2+0+3+(-1)+4+0+0=9$, maksymalny możliwy przepływ po tej drodze wynosi oczywiście $F=1$. Przesyłamy 1 jednostkę i otrzymujemy przepływ o całkowitej wartości $F=1+1=2$, czyli tyle ile chcieliśmy otrzymać, tzn. $F=K=2$. Skumulowane wartości przepływów łukowych wynoszą teraz: $f_{s3'} = f_{3'3''} = f_{3'5'} = f_{5'5''}=1, f_{5''4'} = 0, f_{4'4''} = 1, f_{4''7'} = 0, f_{7'7''}= f_{7''t} = f_{s1'} = f_{1'1''} = f_{1''4'} = 1, f_{5''8'} = f_{8'8''} = f_{8''7'} = 1, f_{4''6''} = f_{6''6''} = f_{6''t} = 1$. Należy zauważyć, że z łuków którymi w obu iteracjach przesyłaliśmy jakkolwiek przepływ, $f_{5''4'}=0$ oraz $f_{4''7'}=0$, gdyż w pierwszej iteracji przesyłaliśmy tymi łukami przepływ równy 1, ale w drugiej iteracji przesyłaliśmy łukami przeciwnymi również przepływ równy 1, czyli tak jak byśmy przesyłali w tym samym kierunku przepływ równy $1+(-1)=0$.

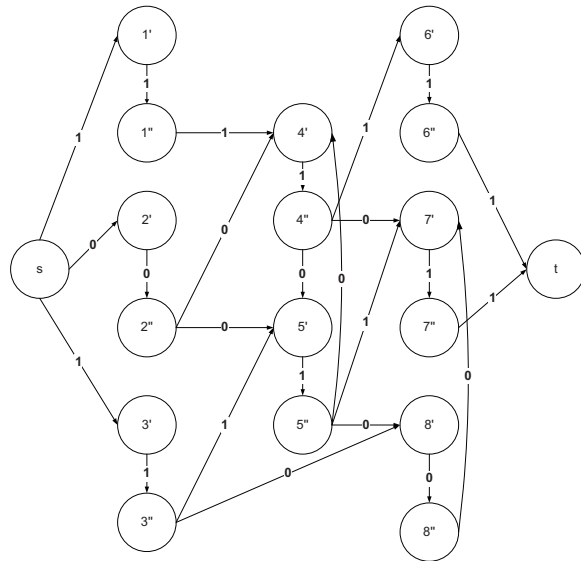


Rys.3. Sieć \bar{S} po pierwszej iteracji wraz ze zmodyfikowanymi kosztami \bar{t}_{ij} oraz przepustowościami \bar{c}_{ij}

Na Rys.4 pokazano sieć S^{**} po iteracji nr 2 (ostatniej) oraz po przypisaniu łukom sieci ostatecznych wartości przepływu otrzymanych po realizacji algorytmu. Wartości przepływów łukowych otrzymane po zakończeniu działania algorytmu identyfikują nam 2 drogi rozłączne wierzchołkowo z s do t :

- ♦ $s \rightarrow 1' \rightarrow 1'' \rightarrow 4' \rightarrow 4'' \rightarrow 6' \rightarrow 6'' \rightarrow t$;
koszt (czas przejazdu) wynosi:
 $0+0+2+0+4+0+0=6$;

- ♦ $s \rightarrow 3' \rightarrow 3'' \rightarrow 5' \rightarrow 5'' \rightarrow 7' \rightarrow 7'' \rightarrow t$;
koszt (czas przejazdu) wynosi:
 $0+0+1+0+8+0+0=9$;



Rys.4. Sieć S^{**} po zakończeniu działania algorytmu z opisanymi na łukach przepływami końcowymi

Na tej podstawie możemy w prosty sposób otrzymać zbiory I_k oraz T_k z (7) i (8) przechodząc jednocześnie na sieć wyjściową S z Rys.2a. Otrzymamy:

$$I_1 = (1,4,6)$$

$$T_1 = (0,2,2+4) = (0,2,6);$$

$$I_2 = (3,5,7)$$

$$T_2 = (0,1,1+8) = (0,1,9); \blacksquare$$

Złożoność algorytmu Busackera-Gowena wynosi [1]: $n_{\max} \cdot O(V^2)$, gdzie

$$n_{\max} = \min\{(Deg(s)-1), (Deg(t)-1), f_{\max}\} \quad (35)$$

przy czym $Deg(s)$, $Deg(t)$ oznaczają stopień wierzchołka źródłowego s oraz docelowego t , f_{\max} oznacza maksymalną wartość przepływu, którą chcemy osiągnąć. Biorąc pod uwagę, że maksymalna wartość przepływu, którą chcemy osiągnąć jest równa K otrzymujemy złożoność algorytmu dla naszego problemu równą $K \cdot O(V^2)$. Jeżeli jako algorytmu wyznaczania najkrótszej drogi użyjemy efektywnej implementacji algorytmu Dijkstry ([6], [9], [11], [17], [39]) opartej o kopce Fibonacciego ($O(V \log V + A)$ zamiast $O(V^2)$), wówczas złożoność obliczeniowa rozpatrywanego problemu wynosi $K \cdot O(V \log V + A)$, gdzie $V=2W+2$ oraz $A=U+W+K_s'+K_d'$. Zauważmy, że

z założenia $i^s \cap i^d = \emptyset$ wynika, iż $K'_s \leq W/2$ oraz $K'_d \leq W/2$ (pamiętamy, że W oznacza liczbę wierzchołków w sieci wyjściowej S). Odnosząc zatem złożoność opisywanego algorytmu do rozmiaru pierwotnej sieci S (S') możemy napisać, iż złożoność prezentowanego algorytmu wynosi $K \cdot O(W \log W + U + W)$. Powstaje jednakże problem związany z tym, że algorytm Dijkstry nie działa poprawnie jeżeli wagi łuków są liczbami ujemnymi (a u nas tak będzie w sieci zmodyfikowanej \bar{S} , zob. Rys.3). Można wówczas wykorzystać modyfikację algorytmu Dijkstry zaproponowaną przez Edmonsa i Karpa w [13]. Zaproponowali oni, aby po każdej modyfikacji sieci \bar{S} znaleźć najmniejszy z ujemnych kosztów łukowych t_{min} sieci i następnie do kosztu każdego łuku dodać liczbę $-t_{min}$. Pozbywamy się wówczas ujemnych kosztów sieci, a rozwiązanie (w sensie drogi, a nie jej długości !) jakie otrzymujemy w sieci \bar{S} po zwiększeniu kosztów wszystkich łuków o $-t_{min}$ jest takie samo, jak w sieci \bar{S} z ujemnymi kosztami.

6. Rozszerzenia problemu planowania manewru

Rozpatrzmy teraz kilka rozszerzeń rozważanego problemu.

(A) Przypadek manewru niejednorodnych obiektów

Pozostaniemy przy problemie (18)-(19) z ograniczeniami (21). W poprzednim punkcie podano metodę znalezienia rozwiązania optymalnego tego problemu, przy założeniu (28). Zakładaliśmy zatem, że przemieszczane obiekty są jednorodne z punktu widzenia czasu pokonywania tego samego odcinka drogi (łuku), tzn., że przemieszczamy identyczne obiekty. Podamy teraz sposób pozbycia się tego ograniczenia. Zauważmy przy tym, że algorytm opisany w poprzednim punkcie działa w ten sposób, że dokładnie K razy szukana jest droga najkrótsza z s do t w zmodyfikowanej na bieżąco sieci \bar{S} . Z (3) wynika, że czasy pokonania łuku (i,j) dla k -tego obiektu zależą od maksymalnej prędkości v_k tego obiektu. Uporządkujmy zatem obiekty manewrowe zgodnie z ich rosnącymi prędkościami poruszania się, tzn. przypiszmy im takie kolejne numery $1, \dots, K$, że zachodzi :

$$v_1 \geq \dots \geq v_k \geq \dots \geq v_K \quad (36)$$

Wówczas otrzymamy następującą zależność (por. z (3)) :

$$\forall_{k=1, K-1} \quad \forall_{\substack{n=2, K \\ n > k}} \quad \forall_{(i,j) \in U_G} \quad t_{ij}^k \leq t_{ij}^n \quad (37)$$

Zależność ta mówi, że dla wybranego łuku (i,j) czas pokonania tego łuku przez obiekt o numerze $k+1$ będzie dłuższy niż przez obiekt o numerze k , $k=1, \dots, K-1$. Rozwiązanie problemu (18)-(19) z ograniczeniami (21) dla K dróg wierzchołkowo rozłącznych może być następujące: w k -tej iteracji algorytmu Busackera-Gowena będziemy wyznaczać najkrótszą drogę z s do t dla k -tego obiektu uporządkowanego zgodnie z (36). W związku z tym, w zmodyfikowanej sieci \bar{S} kosztem \bar{t}_{ij} łuku (i,j) w k -tej iteracji będzie liczba \bar{t}_{ij}^k z zastrzeżeniem, że jeżeli w sieci z iteracji $k'=k-1$, która używana jest w iteracji bieżącej, jakiś łuk został wyrugowany (tzn. $\bar{t}_{ij} = \infty$, zob. (34)), to oczywiście musimy również przypisać temu łukowi $\bar{t}_{ij} = \infty$. Jeżeli z kolei jakiemuś łukowi (j,i) w iteracji k' został przypisany koszt $\bar{t}_{ji} = -\bar{t}_{ij}^{k'}$ (zob. (31)), to w bieżącej iteracji również musimy przypisać temu łukowi wartość $\bar{t}_{ji} = -\bar{t}_{ij}^k$. Wartości przepływów pozostawiamy takie, jakie otrzymujemy w kolejnych iteracjach.

(B) Przypadek dla dróg nierozłącznych (kolizyjnych)

Najprostszym przypadkiem dla dróg kolizyjnych są drogi rozłączne łukowo. W takich drogach mogą powtarzać się jedynie wierzchołki dla obiektów. W związku z tym istnieje niebezpieczeństwo, że w tym samym czasie więcej niż jeden obiekt przejeżdżałby przez jakiś wierzchołek. Rozpatrywanie tego przypadku szczegółowo przekracza zakres tego opracowania, wskażemy zatem jedynie ideę rozwiązania tego problemu. Problem kolizyjnych dróg opisany został między innymi w [4], [16], [22]. Rozpatrywany może być nie tylko dla wspólnych wierzchołków, ale również dla wspólnych odcinków dróg (łuków sieci). Wówczas problem ten staje się nietrywialny i złożony, i można próbować szukać jedynie metod dających przybliżone rozwiązania. Wracając do dróg rozłącznych łukowo, wystarczy zastosować algorytm Busackera-Gowena w sieci S z Rys.2, powiększając ją o wierzchołek s i t i dodając odpowiednie łuki do lub z tych wierzchołków (tak jak przy tworzeniu sieci S^* z S w punkcie 3 opisu). W algorytmie znajdowania najkrótszej drogi należałoby

uwzględnić okresy zajętości poszczególnych wierzchołków, wynikające z wyznaczenia dróg dla wcześniejszych obiektów i przy takiej modyfikacji przeprowadzać realizację wyżej wymienionego algorytmu. Wydaje się, że ten problem można w stosunkowo łatwy sposób rozwiązać.

(C) *Inne rozszerzenia*

Następne ograniczenia dla rozważanego problemu mogą wynikać z faktu, że drogi dla poszczególnych obiektów muszą spełniać pewne dodatkowe warunki (np. na minimalne i maksymalne odległości między nimi, muszą przechodzić przez pewne zadane wierzchołki pośrednie, itp.). Rozwiązania tych i temu podobnych problemów można znaleźć np. w [4], [12], [14]. Problem przechodzenia dróg przez wybrane wierzchołki pośrednie mógłby być zastosowany u nas np. do tego, aby przemieszczać K obiektów z ustalonych wierzchołków początkowych do ustalonych wierzchołków końcowych. Tak rozumiany problem wyznaczania dróg rozłącznych jest znanym problemem trudnym obliczeniowo (należy do klasy NP) i opisywany jest m.in. w [25], [26], [30]. Dla przykładu, w pracy [30] opisano aproksymacyjny algorytm znajdowania K rozłącznych dróg, który jest uogólnieniem algorytmu Dijkstry. Rozwiązanie problemu przemieszczania obiektów z rejonu początkowego do końcowego przedstawione w tym opracowaniu nie zapewnia bowiem przemieszczania tych obiektów ze ściśle określonych wierzchołków do innych konkretnych wierzchołków. Nawet wybranie dokładnie K wierzchołków początkowych i końcowych (po jednej parze dla każdego obiektu) nie zapewnia nam tego, że każdym obiektem będziemy się przemieszczać do ustalonego wierzchołka końcowego. Otrzymane rozwiązanie gwarantuje nam jedynie tyle, że wyznaczymy dokładnie K dróg rozłącznych wierzchołkowo (jeśli tyle oczywiście istnieje), dotrzemy wszystkimi obiektami do rejonu końcowego (zbioru wierzchołków), przy czym każdy obiekt osiągnie inny wierzchołek z tego rejonu (ale nie mamy wpływu na to, który to będzie wierzchołek z podzbioru wierzchołków końcowych) i suma czasów przemieszczania wszystkich obiektów będzie najmniejsza z możliwych. Kolejne rozszerzenia tego problemu mogą dotyczyć sytuacji oddziaływania na sieć przez potencjalnego przeciwnika. Powstaje wówczas problem planowania manewru w warunkach stochastycznych. Rozwiązanie podobnego problemu zostało zaproponowane np. w [7], [28].

7. Podsumowanie

Rozpatrywany w artykule problem wraz przedstawionymi w punkcie 6 rozszerzeniami wpisuje się w etap planowania manewru. Na etapie symulacji przemieszczania (np. kolumn marszowych) istnieje potrzeba synchronizacji przemieszczanych kolumn [29]. Rozwiązuje się wówczas problem sterowania optymalnego (ruchem kolumn) celem np. zsynchronizowania czasów ich dotarcia do wybranych punktów synchronizacji [35]. Innym, nieopisywanym w punkcie 6 rozszerzeniem rozważanego w artykule problemu może być problem planowania manewru (poszukiwania dróg) w warunkach wielokryterialnych. Obszerny przegląd stosowanych rozwiązań oraz adaptacji standardowych algorytmów wyznaczania dróg najkrótszych do rozwiązania problemu wielokryterialnego przedstawiono w [34], [38].

8. Bibliografia

- [1] Ahuja R.K., Magnanti T.L., Orlin J.B.: *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs 1993.
- [2] Benton J.R., Iyengar S.S., Deng W., Brenner N., Subrahmanian V.S.: Tactical route planning: new algorithms for decomposing the map, *Proceedings of the IEEE International Conference on Tools for AI*, 6-8 November, Herndon 1995, pp. 268-277.
- [3] Busacker R.G., Gowen P.J.: A procedure for determining a family of minimum-cost flow patterns, *Operations Research Office Technical Report 15*, John Hopkins University, Baltimore 1961.
- [4] Cai X., Kloks T., Wong C.K.: Time-varying shortest path problems with constraints, *Networks* **29** (1997), pp.141-149.
- [5] Campbell C., Hull R., Root E., Jackson L.: Route planning in CCTT, in *Proceedings of the 5th Conference on Computer Generated Forces and Behavioural Representation*, Technical Report, Institute for Simulation and Training, pp. 233-244, 1995.
- [6] Cormen T.H., Leiserson Ch.E., Rivest R.L.: *Wprowadzenie do algorytmów*, WNT, Warszawa 1997.
- [7] Cormican K., Morton D., Wood K.: Stochastic network interdiction, *Operations Research* **46** (1998), pp. 184-197.
- [8] *Corps Battle Simulation (CBS), Version 1.6.0, Analyst's Guide, Volume 1, Ground*, U.S. Army Simulation, Training, and

- Instrumentation Command Orlando, Florida, June 2001.
- [9] Deo N.: Teoria grafów i jej zastosowanie w technice i informatyce, WNT, Warszawa 1980.
- [10] Deo N., Sysło M., Kowalik J.: *Algorytmy optymalizacji dyskretnej*, wyd. II, PWN, Warszawa 1995.
- [11] Dijkstra E.: A note on two problems in connection with graphs, *Numerische Mathematik* **1** (1959), 269-271.
- [12] Dreyfus S. E.: An appraisal of some shortest path algorithms, *Operations Research* **17** (1969), pp. 395-412.
- [13] Edmonds J., Karp R.M.: Theoretical improvement in algorithmic efficiency for network flow problems, *J. ACM* **19** (1972), pp. 248-264.
- [14] Golden B.L., Skiscim C.C.: Solving k-shortest and constrained shortest path problems efficiently, *Network Optimization and Applications* **20** (1989), Texas A&M University, College Station.
- [15] James J., Sayrs B., Benton J., Subrahmanian V.S.: Uncertainty management: keeping battlespace visualization honest, *Proceedings of the 3rd Annual Conference on Advanced Telecommunications and Information Distribution Research Program (ATIRP)*, University of Maryland, College Park, MD, February 1-5, 1999.
- [16] Kaszubowski Z., Mizera R., Piasecki S.: *Problemy przegrupowania wojsk*. WAT, Warszawa 1970.
- [17] Korzan B.: Elementy teorii grafów i sieci, Metody i zastosowania. WNT, Warszawa 1978.
- [18] Kreitzberg T., Barragy T., Nevin B.: Tactical movement analyzer: a battlefield mobility tool, *Proceedings of the 4th Joint Tactical Fusion Symposium*, Laurel, 1990.
- [19] Longtin M., Megherbi D.: Concealed routes in ModSAF, in *Proceedings of the 5th Conference on Computer Generated Forces and Behavioural Representation*, Technical Report, Institute for Simulation and Training, pp. 305-314, 1995
- [20] Minieka E. : On computing sets of shortest path in a graph, *Comm. Assoc. Comput. Mach.* **17** (1974), 351-353.
- [21] Mitchell J.S.B.: Geometric shortest paths and network optimization, in J.R. Sack and J. Urrutia: *Handbook of Computational Geometry*, Elsevier Science Publishers, B.V. North-Holland, Amsterdam 1999.
- [22] Mizera R.: Teoria badań operacji. Algorytmy planowania transportu, WAT, Warszawa 1972.
- [23] Petty M.D.: Computer generated forces in Distributed Interactive Simulation, *Proceedings of the Conference on Distributed Interactive Simulation Systems for Simulation and Training in the Aerospace Environment*, 19-20 April, Orlando (USA) (1995), 251-280.
- [24] Rajput S., Karr C.: Unit Route Planning, *Technical Report IST-TR-94-42*, Institute for Simulation and Training, Orlando (USA) (1994).
- [25] Schrijver A., Seymour P.: Disjoint paths in a planar graph – a general theorem. *SIAM Journal of Discrete Mathematics* **5** (1992), 112-116.
- [26] Schrijver A.: *Combinatorial Optimization. Polyhedra and Efficiency*, Springer-Verlag, Berlin - New York (2004).
- [27] Software Engineer Maintenance Manual, Volume I. Model Overview and interface Specification, Joint Theater Level Simulation (JTLS), Defense Information Systems Program, Jet Propulsion Laboratory, Pasadena, California, September 1985.
- [28] Tarapata Z.: Multi-paths optimization in an unreliable time-dependent networks, *Proceedings of The 2nd NATO Regional Conference on Military Communication and Information Systems*, 04-06 October, Zegrze (Poland) 2000, vol.I, 181-189.
- [29] Tarapata Z.: Modelling, optimisation and simulation of groups movement according to group pattern in multiresolution terrain-based grid network, *Proceedings of The 3rd NATO Regional Conference on Military Communication and Information Systems*, 10-12 October, Zegrze (Poland) 2001, vol.I, 241-251.
- [30] Tarapata Z.: Algorithm for simultaneous finding a few independent shortest paths, *Conference Proceedings of the 9th European Simulation Symposium*, Passau (Germany) 1997, pp. 89-93.
- [31] Tarapata Z.: Modelling of terrain for necessities of military objects movement simulation, *Bulletin of Military University of Technology*, N^{oo} 1 (2000), pp. 127-146.
- [32] Tarapata Z.: Military route planning in battlefield simulation: effectiveness problems and potential solutions, *Journal of Telecommunications and Information Technology*, N^{oo} 4 (2003), pp. 47-56.

- [33] Tarapata Z.: Models and methods of movement planning and simulation in simulation aided system for operational training, *Proceedings of The 6th NATO Regional Conference on Military Communication and Information Systems*, ISBN 83-920120-0-3, 06-08 October, Zegrze (Poland) 2004, pp. 152-161.
- [34] Tarapata Z.: Selected multicriteria shortest path problems: an analysis of complexity, models and adaptation of standard algorithms, *International Journal of Applied Mathematics and Computer Science*, Vol.17, No.2 (2007), 269-287.
- [35] Tarapata Z.: Automatization of decision processes in conflict situations: modelling, simulation and optimization, w: Arreguin J.M.R. (edt.): *Automation and Robotics*, ISBN 978-3-902613-41-7, I-Tech Education and Publishing, Vienna (Austria) 2008, 297-328.
- [36] The Analyst Guide, Joint Theater Level Simulation (JTLS), Version 1.65, Modern Aids to Planning Program (MAPP), Force Structure and Assessment Directorate (J-8), Joint Staff, The Pentagon, September 1988.
- [37] Wagner R. A.: A shortest path algorithm for edge-sparse graphs, *Journal Assoc. Comp. Mach.* 23 (1976), 50-57.
- [38] Warburton A.: Approximation of Pareto optima in multiple-objective, shortest-path problems, *Operations Research* 35 (1987), pp. 70-79.
- [39] Wilson R.J.: *Wprowadzenie do teorii grafów*, PWN, Warszawa 1998.